

Amendment to the Drawings:

The attached drawing sheet labeled Figure 1 includes changes to Figure 1, specifically the notation of “Prior Art” as required by the Examiner. This sheet replaces the original drawing sheet containing Figure 1. In response to the Examiner’s requirement, a completely new set of drawing sheets for Figures 2 – 5 is also attached. These drawing sheets replace the original drawing sheets containing Figures 2 – 5.

Attachment: Replacement Sheets

REMARKS/ARGUMENTS

INTERVIEW SUMMARY

During a very productive discussion with the Examiner, several key points of the invention were discussed which have a direct bearing on the claims. Before proceeding to respond to the issues raised by the Examiner in the Office Action dated July 29, 2005, Applicants will summarize these points since they will serve as the basis for the Response.

Initially Applicants wanted to clarify the definition of “object” since the Examiner had raised the issue in the Office Action and an understanding of “object” is central to the understanding of Applicants’ invention. The Examiner’s attention was drawn to the definition of “object” contained in the specification. Applicants submit that within the definition provided by the specification, those skilled in the art will understand an object to be an encapsulated executable which can be instantiated and operate on supplied parameters. The specification discusses the use of objects written to some known specification and gives an example of VBX objects.

During the interview, Applicants pointed out that at the time of Applicants’ invention, the prior art taught that objects could, during their construction, have written into them any number of properties and events, and it was common practice at that time for developers to use one or more class libraries to create a set of compatible objects. However, it was not appreciated how to add additional properties and events to a third party objects without rewriting the objects. The invention of the present application adds properties and events to existing third party objects without rewriting the objects. As disclosed in the application, the addition of properties and

events is accomplished by wrapping the objects within instances of a wrapping object .

A second feature of applicants' invention that was discussed is the dynamic (real time) wrapping of objects with additional properties and events by both the development and the runtime program as the programs are running. Initially, the objects with their built-in (native) properties and events reside in storage locations accessible to the programs of the invention. At this point, the objects have no additional properties and events added. Then, when the development and runtime programs of the invention require an object, the object is queried to determine its native properties and events, and the object is dynamically wrapped to add the additional properties and events that are used to control the object. In the exemplary program disclosed in the application, a single set of additional properties and events are added to each wrapped object. However, as discussed with the Examiner, different objects may be wrapped with different properties and events if that is required. The wrapping is a dynamic process and is accomplished on-the-fly. No permanent repository of the wrapped objects is kept.

Also discussed was an aspect of the invention that results from the objects being dynamically wrapped, namely the fact that the objects can be located anywhere as long as the location of the object is provided to the program (either development or runtime) and the object can be accessed. The script contains not only a listing of properties and events, but also contains the storage location for each object used by the script. The script is a separate file that has its own location. The script's location can be provided to the program by several methods well known in the art, for example by a command line argument, or by another script, or by dragging and dropping. Similar to objects, the script may also be located anywhere as long as the location of the script is provided to the program (either development or runtime) and the script can be

accessed. Consequently, it is not necessary for the development program, the runtime program, the scripts, or the objects to be physically located near one another. It was pointed out to the Examiner that this is the basis for the “maintained separately” language relating to the objects and scripts recited in the claims.

Finally, the Applicants discussed with the Examiner the generality achieved by the construction of Applicants’ invention. The invention places no limits upon the functions contained within the objects which are used and no limits on the abilities of the objects to interact with systems outside the invention’s programs. For instance, as one type of example, it was pointed out that the program implements logical operators and looping constructions, variables, conditionals, and other features by employing objects which perform those functions. Similarly, objects implementing any other imaginable function can be used with the invention. Any type of program and/or program construct can be formulated with Applicants’ invention. Examples of parallel processing programs, robotics, and language translation programs as being possible to implement with Applicants’ invention were mentioned during the interview, but are by no means meant to limit the generality of the possible uses of Applicants’ invention.

The general features of the invention discussed above are reflected in the claims. During the interview, Applicants provided the Examiner, at the Examiner’s request, with a DVD version of the VHS video tape that had been in the Patent Office file of the parent application (issued as U.S. Patent No. 5,862,372) but which could not be located by the Examiner. The VHS video tape had been previously supplied for review as an aid in helping the examiner of the parent application understand the program. In addition, Applicants briefly displayed a working/running version of the program and showed the Examiner how programs were constructed with the

invention.

During an earlier telephonic interview and again at the October Interview, Applicants' attorney pointed out to the Examiner that Applicants had become aware that the software code appendix supplied on CD-ROM as part of the application was missing a single header file (workform.h) from the initial parent application program. (The header file is listed in the file list contained on the CD-ROM, but, for reasons unknown to the inventors, the actual file failed to copy to the CD-ROM during duplication). In the parent application, the entire code had been supplied as a several hundred page printout. During prosecution of the parent, Applicants were requested to file the code as a CD-ROM appendix and did so. A copy of that CD-ROM was filed with the present application. The header file that is missing from the CD-ROM does not affect the scope of the claims or the disclosure. This file is necessary to compile the program (since there is an internal reference to the file) but it can be easily generated by those skilled in the art from information contained within the program code without undue experimentation. Applicants submit that the code provided on CD-ROM provides a sufficiently complete written description of the invention and is enabling of the claims. The Examiner indicated that he understood what had happened and that he did not believe it presented any written description or enablement problem for the application.

The paper copy printout of the entire program code is still maintained in the original application. Attached to this Response is a copy of the missing header file copied from the program code printout submitted in the parent application. It is placed in the record of this application as a convenience to anyone desiring it who does not wish to go to the effort of constructing the header file themselves.

Applicants will address all issues raised by the Examiner in the Office Action following the Examiner's paragraph order.

Examiner's Paragraph 3:

The Examiner has required a replacement Declaration complying with the most recent 1.56 requirements. The required Declaration is attached to this Response.

Examiner's Paragraph 5:

A requirement to place a label on Figure 1 designating it as Prior Art has been imposed. A corrected drawing replacement sheet is attached to this Response.

Examiner's Paragraph 6:

The Examiner has noted a PTO Draftsperson objection to Figures 2 – 6 (shading too dark in screen captures) dated December 9, 2001 and has required corrective action.

Applicants' Response:

Applicants note that the Figures submitted with this application were copies of the original figures submitted in the parent application that were found suitable for publication. The copying process used by Applicants most likely filled in some of the backgrounds in the Figures resulting in the Draftsperson's objection. Applicants understand the Examiner's comments about the shading in the screen captures being too dark to reflect this background fill. In response to the Examiner's requirement, attached are newly printed replacement sheets of the Figures.

Examiner's Paragraph 7:

Claim Rejections - 35 USC § 112

The Examiner has rejected claims 1 - 11, 16 - 18, 20 - 38, 43 - 45, and 47 - 57 under 35 U.S.C. 112, first paragraph:

“...because the specification, while being enabling for the wrapping of objects in a Development Environment, does not reasonably provide enablement for a broad interpretation of inventing wrappers in the field of Object technology. The specification does not enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to the advancement of wrappers in object technology the invention commensurate in scope with these claims. For Example, the scope of Claim 2 is the field of object technology. The scope of dependent claim 3 is the field of object oriented icons which are used during application development playback. The broad scope of claim 2 with the scope of claim 3 does not provide sufficient support for object oriented icons providing application development playback outside of the unclaimed development environment.”

Applicants' Response:

Applicants make no claim to having invented wrapping in the field of Object technology. This point was discussed with the Examiner during the interview. One feature that Applicants do claim to have invented is a method for adding additional properties and events to objects that have been written to a known specification. Such objects have their own native/internal properties and events that were included as they were written. Applicants' invention dynamically wraps those objects, written to a known specification, within another object and thereby causes the objects written to a known specification to respond to the additional properties and events. This dynamic wrapping takes place in both the development and run time programs.

A second feature of Applicants' invention arises from: 1) the fact that Applicants' run time program is interpreted and not compiled; and 2) the fact that the wrapping with additional properties and events is done dynamically by the run time interpreter. Therefore, the objects and the script containing a list of properties and events can be stored in any location accessible to the run time program such as, but not limited to, local hard drives, local area networks, or internet locations.

Applicants submit that the specification is fully enabling of the above referenced features of the invention including the dynamic wrapping of objects in both the development and run time applications. Applicants have amended the claims to more succinctly recite their invention. With the exception of claims 20 and 47, Applicants have refocused the claims to primarily recite the run time feature. In addition, Applicants have amended the claims to recite: 1) the dynamic wrapping of objects; and 2) that the objects are written to a known specification. Also to add greater clarity, the claims now recite "one or more objects" and "one or more scripts." As the claims were rewritten to focus on the run time aspect, several claims became redundant and have been canceled. In addition, claims 2 and 3 and 29 and 30 have been canceled for reasons not relating to the Examiner's rejection. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Examiner's Paragraph 8:

Claim Objections

The Examiner has object to claims 1, 2, 4, 8, 17, 18, 20 - 23, 25, 26 - 29, 31, 35, 44, 46 - 50

and 52 - 57 due to the presence of more than one period in the claims, specifically those periods following the subsection indicators. Correction has been required.

Applicants' Response:

The claims have been amended to replace the period after each subsection identifier with a parenthesis. Applicants submit that with this amendment, the basis for the Examiner's objection has been overcome, and Applicants respectfully request that the Examiner remove the objection of record.

Examiner's Paragraph 10:

Claim Rejections - 35 USC § 112 second paragraph

The Examiner has rejected claims 26 and 53 under 35 U.S.C. § 112, second paragraph, as:

"...being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Examiner finds the term "at least partially concurrently" to be indefinite. Although, familiar with concurrent it is not clear what "at least partially concurrently" means."

Applicants' Response:

Applicants have amended claims 26 and 53 to remove the recitation of "at least partially." Applicants submit that with this amendment, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Examiner's Paragraph 11:

The Examiner has rejected claims 16 and 31 under 35 U.S.C. § 112, second paragraph, for:

having an indefinite limitation. The limitation “wherein unlimited expansion of program capabilities is achieved.” is not definite. Computers are finite machines with limited amounts of storage. It is unclear what the limitation means.

Applicants’ Response:

Applicants believe that the Examiner had intended to indicate claims 16 and 43 both of which contain the “unlimited expansion” language. Applicants have taught that any object written to a known specification can be utilized by their method. As examples, objects implementing such features as program constructs or sub-programs may be employed. The important feature of Applicants’ invention is that there is no limitation imposed by Applicants’ invention on the type of function an object may perform. Accordingly, the expansion of program capabilities in Applicants’ invention is only limited by the ingenuity developers have in creating new objects. This is the meaning Applicants intend by the claim phrase “wherein unlimited expansion of program capabilities is achieved.” In the context of Applicants invention, Applicants respectfully submit that the phrase “wherein unlimited expansion of program capabilities is achieved” is definite and does particularly point out and distinctly claim the subject matter which applicant regards as the invention. Nonetheless, Applicants appreciate the Examiner’s concern with the use of the term “unlimited” and accordingly Applicants have amended the claim to eliminate that term. However, Applicants believe that the claim still recites the intended meaning, namely that the expansion of program capabilities is only limited by the availability of different objects. Accordingly, Applicants respectfully request that the Examiner remove the rejections of record.

Examiner's Paragraph 13:

Claim Rejections - 35 USC § 103

The Examiner has rejected claims 1,2,4,8,16 - 18,23,24,28,30,31,35 - 38,43 - 44 and 51 - 56 under 35 U.S.C. 103(a) as being obvious and unpatentable:

"...over "The Xerox **Star**": A Retrospective, by Jeff Johnson et al published September 1989, IEEE Computer in view of USPN #5,6 19,637 Henshaw et al filed December 2, 1993 (referred to as **Wrapper**)."

Claim 1

A computer implemented system employing objects for generating an application script, in which both the objects and the script may can be maintained separately, comprising:

- a. means for wrapping objects with additional properties and events beyond those properties and events internal to the object; and
- b. means for utilizing the additional properties and events to link and sequence the objects..

Examiner's Response

Star teaches generating an application script (Star, page 4, transferred into documents" - the text created for objects). Star also on page 4 mentions the ability to edit in Star and the format is text files (as in residing on disk in a directory in a text file able to be edited by a separate word processor). Furthermore, the mention of editors such as Word which can be used to edit text files separate of the Star environment. Star does not explicitly teach the wrapping of objects. It is Wrapper who uses container objects to wrap objects within the graphical user interface (Wrapper, Abstract). An object by definition is attributes and the methods to perform operations on those attributes. The limitations of "with additional properties (attributes) and events (methods

which produce messages) is met by the use of container objects. The content of the container object (holding an object or objects, Wrapper, ccl. 1, lines 50 - 64) may have the content displayed (Col 2, lines 1 — 20) and sequence arranged (Wrapper, Figure 5, col 5, lines 20— 33 and Star Figure 4, page 11). Star teaches a development environment with icons which are text based and able to be manipulated internal to Star or by editing the text file by word processor and Wrapper teaches a means of wrapping objects in a container object there by adding attributes and methods. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine Star and Wrapper because, wrapped objects enable iconic programming in a Graphical User Interface environment.

Applicants' Response:

Applicants respectfully disagree with the Examiner's interpretation of the cited prior art and in support thereof, present the following arguments. "The Xerox Star" as described in the retrospective by Jeff Johnson does not make obvious the current invention for a number of reasons. The article on the Star system clearly states that the Star system is a Desktop metaphor designed for office workers, "Star's users would be casual, occasional users" and that:

"The Desktop metaphor was intended for office automation and publishing. It may **not** (emphasis added) be appropriate for other applications (e.g., software development).

However, it can be argued that orienting users toward their data rather than toward application programs and employing analogies with the physical world are useful techniques in any domain." (page 5, paragraph 4)

Further, Star's intent was to provide a working environment (desktop) for documents (text manipulation):

"The document is the heart of the world, and unifies it -- Most personal computers and workstations give no special status to any particular application. Dozens of applications are available, most of which are incompatible with each other in data format as well as user interface.

Star, in contrast, assumes that the primary use of the system is to create and maintain documents. The document editor is thus the primary application. All other applications exist mainly to provide or manipulate information whose ultimate destination is a document. Thus, most applications are integrated into the document editor (see Integrated applications, above), operating within frames embedded in documents, and those applications that are not part of the document editor support transfer of their data to documents." (page 14, paragraphs 4 and 5)

This data-centric text-centric approach is vastly different from Applicants' inventive program-centric focus. The definition of object in Applicants' invention clearly states that the objects they use are executable programs, not data files. The data files in Star are said to have properties. A set of such properties is associated with an individual data file, and so these properties behave more like initialization parameters found in the old windows .INI files. A text file in Star may have properties like "Author", "font", and "date", but these have no direct effect on the content of the file with which they are associated. While the display of the text may change if the "font" property is set and the text is read into a text display program, the words of the text do not change because the data file has not been directly affected by the property setting, rather the text editor reads the "properties" of the file and changes the display.

There are non-trivial differences between a set of initialization parameters associated

with a single data file, and the type of scripting found in Applicants' invention. Applicants' scripting can be used to control directly the behavior of hundreds of programmable objects at a time, but the differences go beyond that number of objects that can be controlled, instead it lies in the fact that programmable objects are being directly controlled in the first place. In Star there is no direct scripted control of data objects, and executable objects are not even considered. In Star only the user can control the data, and that is done through an associated application program. On the other hand the whole purpose of the Applicants' invention is the direct programmatic control of executable objects. Applicants' invention can even control objects that do not read or use data files.

One thing that is very clear from the Star article is that the Star system is **not** a programming environment, but instead it is an end user workspace environment that is designed to make the work done to be as much like using the physical objects as possible, and to make the computer aspect of the work as minimal as possible. On the other hand the current invention is a programming and application playback environment.

Henshaw's teaching does not combine with the Star system to make the current invention obvious for the following reason. The Henshaw patent covers the subject matter of user interface design, that is; how a user interfaces with the system. When Henshaw discusses putting an object within another object it is referring to moving a first icon into a second folder icon that sits on a virtual desktop space. Henshaw does not teach manipulating "objects" as that term is understood in the art of object oriented programming. The use of the term "object" in Applicants' invention is as an encapsulated executable program, the instantiation, linking, execution, and extinction of which can be effected by the manipulation of properties in a script. Clearly, Henshaw's teaching

of the moving of an icon for a text file into a desktop folder, for example, is a completely different teaching than adding programmable properties to a software object that has its own internal programmable properties which control its behavior. The fact that Henshaw refers to an object does not make it equate to the use of the word object in Applicants' invention or to its use as a term of art in object oriented programming.

In summary, Star and Henshaw combined do not anticipate Applicants' invention. The addition of a non-programming data-centric environment (Star) to a system (Henshaw) that allows the moving of text files into desktop folders does not make obvious a programming and playback system that wraps programmable executable objects with new properties and events that determine the behavior of those programmable objects at runtime. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejections has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 2

A computer implemented system employing objects for generating an application script, in which both the objects and the script can be maintained separately, representing a program structure comprising:

- a. means for simultaneously displaying a plurality of different representations of the program structure; and
- b. means for manipulating the program structure within each of the different representations wherein the representations of the program structure may be synchronized.

Examiner's Response

As per claim 1 and Star Figure 1.

Applicants' Response:

See generally Applicants' response to Examiner's Claim1 rejections (paragraphs 7 and 13.) However, no response is required since this claim has been canceled.

Claim 4

A computer implemented system employing objects and utilizing a script, in which both the objects and the script can be maintained separately, comprising:

- a. a development environment and an interpreting run time environment; and
- b. means for utilizing objects by specifying property values according to the script.

Examiner's Response

As per claim 1 -- The development environment as depicted in the Star reference and the ability to interpret icons in that environment.

Note: no limitations are present to executing the model.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Claim 4 has also been amended to remove the reference to the development environment. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 8

A computer implemented system employing objects and utilizing a script, in which both the

objects and the script can be maintained separately, comprising:

- a. a development environment and an interpreting run time environment that have no logical or **arithmetic operators**; and
- b. means for utilizing objects by specifying property values according to the script.

Examiner's Response

Bold indicates the limitation met by the rejection in view of the OR.

As per claim 1 - The icon does not require the presence of an arithmetic operators.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Claim 8 has also been amended to remove the reference to the development environment and to now only reference the run time program. Further, the claim has been amended to recite that the scripts contain property settings for the objects. The claim reflects the fact that the run time program of the invention has no logical or arithmetic operators. The system of the invention uses objects having logical or arithmetic operators when such are required. Note that the system uses a run time program, objects, and scripts which can be maintained separately. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 16

The system of claim 4, 5, 6, 7, 8, 9, 10, or 11 further comprising a means for adding additional programming constructs by employing objects that perform the function of programming constructs wherein unlimited expansion of program capabilities is achieved.

Examiner's Response

Bold indicates the claim the current limitation is dependent on for purposes of rejection. As per the rejection of claim 1.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) The claims from which claim 16 depends have been amended to remove the reference to the development environment and to now only reference the run time program. The claim reflects the fact that programming constructs or even sub-programs can be added to the system of the invention by employing objects which performs those functions. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 17

A computer implemented system employing objects and interpreting a script in which both the objects and the script may can be maintained separately, comprising:

- a. a run time program; and
- b. means for utilizing objects according to the script.

Examiner's Response

As per claim 1. The Star system is a runtime program.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's claim 1 rejections

(paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 18

A computer implemented system employing objects and interpreting a script in which both the objects and the script may can be maintained separately, comprising:

- a. a run time program that has neither **arithmetic** nor logical operators; and
- b. means for utilizing objects according to the script.

Examiner's Response

As per the rejections for claims 1, 8 and 17

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13) and, in addition, their Response to the Examiner's remarks directed to Claim 8 above. However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 23

A computer implemented run time system employing objects which interprets a script containing property values and event settings, **in which both the objects and the script may** can be maintained separately, and dynamically executes objects comprising:

- a. means for wrapping objects with additional properties and events beyond those properties and events internal to the objects;
- b. means for utilizing the additional properties and events to link and sequence the objects; and c.

means for reading one or more sets of property values and event settings maintained separately from the run time system and the objects wherein the execution of the objects is determined by the property values and event settings in the script.

Examiner's Response

As per claim 1.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 24

The system of claim 23 further comprising means for adding programming constructs or sub-languages utilizing objects.

Examiner's Response

As per claim 1.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 28

A computer implemented software method employing objects for generating an application

script, in which both the objects and the script may can be maintained separately, comprising the steps of:

- a. wrapping objects with additional properties and events beyond those properties and events internal to the object; and
- b. utilizing the additional properties and events to link and sequence the objects.

Examiner's Response

As per claim 1.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 30

The software method of claim 29 further comprising the step of highlighting the icon for each object in the representations as objects are being instantiated during application development run time preview.

Examiner's Response

As per claim 1 and Star page 9 see figure 2.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the

amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 31

A computer implemented software method employing objects and utilizing a script, in which both the objects and the script can be maintained separately, comprising the steps of a. utilizing a development environment and an interpreting run time environment; and b. utilizing objects by specifying property values according to the script.

Examiner's Response

As per the rejection for claim 4.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13) and, in addition, Applicant's Response to the Examiner's Claim 4 rejections. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 35

A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may can be maintained separately, comprising the steps of;

- a. utilizing a development environment and an interpreting run time environment that have no logical or arithmetic operators; and
- b. utilizing objects by specifying property values according to the script.

Examiner's Response

As per the rejection for claim 4.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13) and, in addition, Applicant's Response to the Examiner's Claim 4 rejections. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 36

The software method of claim 35 further comprising the step of communicating among objects through the exchange of property values.

Examiner's Response

As per claim 1. Not interpreted to be executing just establishing the messaging via methods.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 37

The software method of claim 36 further comprising the step of communicating among objects

wherein an event generated by an object triggers an instance of another object.

Examiner's Response

As per claim 1. Not interpreted to be executing just establishing the messaging via methods. Object Oriented technology inherently provides for attributes, methods, messages by definition.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 38

The software method of claim 35 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

Examiner's Response

As per claim 37.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 43

The software method of claim 31, 32, 33, 34, 35, 36, 37, or 38 further comprising the step of adding additional programming constructs by employing objects that perform the function of programming constructs wherein unlimited expansion of program capabilities is achieved.

Examiner's Response

As per claim 16.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) The claims from which claim 43 depends have been amended to remove the reference to the development environment and to now only reference the run time program. The claim reflects the fact that programming constructs or even sub-programs can be added to the system of the invention by employing operators which performs those functions. Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 44

A computer implemented software method employing objects and interpreting a script in which both the objects and the script may can be maintained separately, for executing an application comprising the steps of

- a. utilizing a run time program; and
- b. utilizing objects according to the script.

Examiner's Response

As per claim 17.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 51

The software method of claim 50 further comprising the step of **adding programming constructs** or sub-languages utilizing objects.

Bold indicates the portion of the OR met by the rejection.

Examiner's Response

Adding text as per claim 1 with a wrapper.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 52

A computer implemented software method which interprets a script, which can be maintained separately, containing property values and event settings , which may maintain that distributes

processing to objects, provides and manages data flow among objects, and manages the execution of objects comprising the steps of:

- a. wrapping objects with additional properties and events beyond those properties and events internal to the object; and
- b. utilizing the additional properties and events to link and sequence the objects wherein the execution of the objects is determined by the property values and events.

Examiner's Response

As per claim 1. The distributes processing to objects, provides and manages data flow among objects, and manages the execution of objects is deemed to be claim pre-runtime of the model.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 53

A computer implemented software method employing objects which implements parallel processing comprising the steps of:

- a. wrapping objects with additional properties and events beyond those properties and events provided internal to the object;
- b. utilizing the additional properties and events to link and sequence the objects; and
- c. specifying the temporal relationship among objects by placing the objects on one or more time lines wherein execution of the objects occurs at least partially concurrently and during which property values may be exchanged among the objects and events may be initiated.

Examiner's Response

As per claim 23.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 54

A computer implemented object oriented software programming method in which the function of programming constructs is achieved by dynamically executing objects comprising the steps of:

- a. wrapping objects with additional properties and events beyond those properties and events provided internal to the object;
- b. utilizing the additional properties and events to link and sequence the objects; and
- c. specifying a list of property values and event settings wherein the execution of the objects is determined by the list of property values and event.

Examiner's Response

As per claim 1.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's Claim 1 rejections (paragraphs 7 and 13.) Applicants submit that considering these Remarks and with the amendments, the basis for the Examiner's rejection has been overcome, and Applicants respectfully request that the Examiner remove the rejections of record.

Claim 55

A computer implemented software method providing a general solution for employing standardized objects with properties not internal to the standardized objects comprising the steps of

- a. wrapping standardized objects with additional properties beyond those properties internal to the standardized object; and
- b. utilizing the additional properties to control the standardized objects.

Examiner's Response

As per claim 1 — the standard being the container objects ability to communicate with objects within.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 56

A computer implemented software method providing a general solution for employing standardized objects with events not internal to the standardized objects comprising the steps of:

- a. wrapping standardized objects with additional events beyond those events internal to the standardized object; and

b. utilizing the additional events to control the standardized objects.

Examiner's Response

As per claim 1. Ability to interact with objects not in the container.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Claim 57

A computer implemented software method providing a general solution for employing standardized objects with properties and events not internal to the standardized objects comprising the steps of

- a. wrapping standardized objects with additional properties and events beyond those properties and events internal to the standardized object; and
- b. utilizing the additional properties and events to control the standardized objects.

Examiner's Response

As per claim 55.

Applicants' Response:

Applicants incorporate and repeat herein their response to Examiner's claim 1 rejections (paragraphs 7 and 13.) However, no further response is required since this claim has been canceled due to redundancy with other claims after amendment of those claims.

Examiner's Paragraph 14:

Allowable Subject Matter

The Examiner has allowed claims 20, 21, 22, 25, 26*, 27, 29, 45, 47—50 stating that:

Current prior art of record fails to disclose the claim limitations when taken together or separately. The combined limitations includes instantiating object. Prior art of record does not include the runt time environment to support the instantiating of objects as clearly claimed.

The Examiner allowed claim 26 provisionally provided the Section 112 second paragraph rejection was overcome. Applicants have canceled claims 29 and 45 for reasons having nothing to do with the Examiner's comments and Applicants reserve the right to refile these claims in a subsequent application. Applicants also note that they believe claim 30 which was canceled would have been allowable had claim 29 from which claim 30 depended remained in the application. Applicants submit that with the amendment to claim 26 as discussed above in response to Examiner's paragraph 10, claim 26 is allowable.

Applicants have amended the allowed claims along the same lines as the amendments to all the claims, namely, reciting a dynamic wrapping of objects written to a known specification and one or more objects and one or more scripts. Applicants submit that the allowed claims remain allowable with these amendments and that the amendments serve to more particularly point out that which the applicants believe is their invention.

Examiner's Paragraph 15:

Allowable Subject Matter

The Examiner has objected to claims 3, 5 - 7, 9 - 11 and 32 - 34 as:

...being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. These claims contain the runtime aspect of running the model. The current prior art of record contains the modeling aspect in an interpreted environment.

Applicants respectfully submit that they have addressed and overcome all bases for rejection of the base claims from which claims 5 - 7, 9 - 11 and 32 - 34 depend, and, accordingly, these claims are allowable without rewriting to include the limitations of the base claims. Applicants have canceled claim 3. Applicants also believe that the Examiner intended to include claims 36 - 38 as being allowable if rewritten. Claims 36 - 38 parallel claims 9 - 11. Applicants submit that they have overcome the basis for rejection of claim 35 from which claims 36 - 38 depend and that claims 36 - 38 are, therefore, also allowable without rewriting. Applicants respectfully request that the Examiner allow all these claims except claim 3.

Examiner's Paragraph 16:

Examiner's Observation

The Examiner has suggested that the claims were:

...not entirely directed toward the intended use of the invention as wrapping objects for a Development Environment. They are deemed not commensurate with the scope of

invention and are interpreted in the broadest reasonable interpretation in view of the Specification's support and the scope of claim language in view of the Specification's disclosure of the intended use of the invention. Furthermore, amendment to the preamble of part of the claims could lead to a Restriction. Where part of the claims are directed toward the intent of the invention and part directed toward the field of Object technology. It is the perception of this Examiner this issue has and will impact pendency of the application.

Applicants submit that the clarification with the Examiner during the Interview with respect to the definition of "object" as more than "within the IDE" as noted in the Examiner's Interview Summary overcomes the concerns expressed by the Examiner in Examiner's paragraph 16. Further, Applicants respectfully submit that they have addressed all other aspects of this issue raised by the Examiner in Applicants' answer to Examiner's paragraphs 7 and 13. Applicants have explained that their method of dynamically wrapping objects written to a known standard is utilized both in the development and run time environments. Applicants have amended the claims to focus predominantly on the run time environment. With these amendments Applicants submit that the claims are entirely commensurate with the scope of the invention. Further, Applicants submit that the amendments to the preambles do not raise a basis for restriction.

NEW CLAIMS:

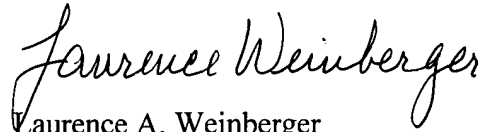
In order to more completely claim and particularly point out that which they consider their invention, Applicants have added new claims 58 –75. No further fees are due.

Accordingly, Applicants respectfully request that the Examiner permit this application to pass to issue.

Serial No. 09/184,738
Amendment dated November 28, 2005
Reply to Office Action dated July 29, 2005

November 28, 2005

Respectfully submitted,



Laurence A. Weinberger
Attorney for Applicants
USPTO Reg. No. 27,965
882 S. Matlack St., Suite 103
West Chester, PA 19382
610-431-1703
610-431-4181 (fax)
larry@lawpatent.com